PTO/SB/17 (01-03)
Approved for use through 04/30/2003. OMB 0651-0032
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

FEE TRANSMITTAL for FY 2005

Effective 10/01/2003. Patent fees are subject to annual revision.

☐ Applicant Claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT ($) 340.00

Complete if Known	
Application Number	09/768,823
Filing Date	January 25, 2001
First Named Inventor	Sexton, et al.
Examiner Name	Cao, D.
Art Unit	2126
Attorney Docket No.	50277-0459

METHOD OF PAYMENT *(check all that apply)*

☐ Check ☒ Credit card ☐ Money Order ☐ Other ☐ None

☐ Deposit Account

Deposit Account Number:

Deposit Account Name:

The Commissioner is authorized to: *(check all that apply)*

☒ Charge fee(s) indicated below ☐ Credit any overpayments

☐ Charge any additional fee(s) during the pendency of this application

☐ Charge fee(s) indicated below, **except for the filing fee**

to the above-identified deposit account.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1001	790	2001	395	Utility filing fee	
1002	350	2002	175	Design filing fee	
1003	550	2003	275	Plant filing fee	
1004	790	2004	395	Reissue filing fee	
1005	160	2005	80	Provisional filing fee	

SUBTOTAL (1) ($)

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

			Extra Claims		Fee from below		Fee Paid
Total Claims		-20**=		X		=	
Independent Claims		- 3**=		X		=	
Multiple Dependent						=	

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
1202	18	2202	9	Claims in excess of 20
1201	88	2201	44	Independent claims in excess of 3
1203	300	2203	150	Multiple dependent claim, if not paid
1204	88	2204	44	**Reissue independent claims over original patent
1205	18	2205	9	**Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($)

*** or number previously paid, if greater; For Reissues, see above*

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1051	130	2051	65	Surcharge – late filing fee or oath	
1052	50	2052	25	Surcharge – late provisional filing fee or cover sheet	
1053	130	1053		Non-English specification	
1812	2,520	1812		For filing a request for *ex parte* reexamination	
1804	920*	1804		Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805		Requesting publication of SIR after Examiner action	
1251	110	2251	55	Extension for reply within first month	
1252	430	2252	215	Extension for reply within second month	
1253	980	2253	490	Extension for reply within third month	
1254	1,530	2254	765	Extension for reply within fourth month	
1255	2,080	2255	1,040	Extension for reply within fifth month	
1401	340	2401	170	Notice of Appeal	
1402	340	2402	170	Filing a brief in support of an appeal	340.00
1403	300	2403	150	Request for oral hearing	
1451	1,510	1451		Petition to institute a public use proceeding	
1452	110	2452	55	Petition to revive – unavoidable	
1453	1,370	2453	685	Petition to revive – unintentional	
1501	1,370	2501	685	Utility issue fee (or reissue)	
1502	490	2502	245	Design issue fee	
1503	660	2503	330	Plant issue fee	
1460	130	1460		Petitions to the Commissioner	
1807	50	1807		Processing fee under 37 CFR 1.17(q)	
1806	180	1806		Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	790	2809	395	Filing a submission after final rejection (37 CFR § 1.129(a))	
1810	790	2810	395	For each additional invention to be examined (37 CFR § 1.129(b))	
1801	790	2801	395	Request for Continued Examination (RCE)	
1802	900	1802		Request for expedited examination of a design application	

Other fee (specify) ____________

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3) ($) 340.00

SUBMITTED BY				Complete *(if applicable)*	
Name (Print/Type)	Margo Livesay, Ph.D.	*Registration No. (Attorney/Agent)*	41,946	*Telephone*	703-425-8516
Signature	Margo Livesay			*Date*	November 19, 2004

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:	Harlan SEXTON et al.		
Application No.:	09/768,823	Group Art Unit:	2126
Filed:	January 25, 2001	Examiner:	Cao, D.
Attorney Docket:	50277-0459	Conf. No.:	2214
Client Docket:	OID-1999-084-01		

For: ACCESSING SHORTER-DURATION INSTANCES OF ACTIVATABLE OBJECTS BASED ON OBJECT REFERENCES STORED IN LONGER-DURATION MEMORY

APPEAL BRIEF

Honorable Commissioner for Patents
Alexandria, VA 22313-1450

Dear Sir:

This Appeal Brief is submitted in support of the Notice of Appeal dated September 17, 2004.

I. REAL PARTY IN INTEREST

Oracle International Corporation is the real party in interest.

II. RELATED APPEALS AND INTERFERENCES

Appellants are unaware of any related appeals and interferences.

III. STATUS OF THE CLAIMS

Claims 1-20 are pending in this appeal. No claim is allowed. This appeal is therefore taken from the final rejection of claims 1-20 on May 17, 2004.

IV. STATUS OF AMENDMENTS

No amendment to claims 1-20 has been filed following the final Office Action of May 17, 2004.

V. SUMMARY OF THE INVENTION

The present invention addresses problems associated with accessing an instance of a recreatable object in a shorter-duration memory based on a reference located in a longer-duration memory. The shorter-duration memory is associated a call. (Specification, p. 3, lines 6-8)

A database system may use a run-time environment for a language such as JAVA, and client processes may establish database sessions with the database system. A database session refers to the establishment of a connection between a client and a database system through which a series of calls may be made. As long as the client remains connected in the database session, the client and the associated database session are referred to as being active. Active clients may submit calls to the database system to request the database system to perform tasks. One example of a call is a query in accordance with the Structured Query Language (SQL), and another example is a method invocation of a JAVA object or class, defined for performing a database task for the database system.

The database system may include, among other components, a database memory for storing information useful for processing calls and a number of server processes for handling individual calls. The database memory may include various memory areas used to store data used

by the server processes. These memory areas may include, for example, a database instance memory, session memories, and call memories. The number of such memories may vary over time as various clients make various calls to the database system.

A database instance memory is a shared memory area for storing data that is shared concurrently by more than one process. For example, a longer-duration memory area may be used store the read-only data and instructions (*e.g.* bytecodes of JAVA classes) that are executed by the server processes. The database instance memory is typically allocated and initialized at boot time of the database system, before clients connect to the database system.

When a database session is created, an area of the database memory is allocated to store information for the database session. For example, session memories may be allocated for multiple clients for each of which a separate database session has been created. A session memory is a shared memory used to store static data, *i.e.*, data associated with a user that is preserved for the duration of a series of calls, especially between calls issued by a client during a single database session. JAVA class variables are one example of such static data.

A call memory may be used to store data that is bounded by the lifetime of a call. When a client submits a call to the database system, one of the server processes is assigned to process the call. For the duration of the call, the server process is allocated a call memory for storing data and other information for use in processing the call. Because the lifetime of objects in a call memory associated with a call is shorter than the lifetime of objects stored in the session memory associated with the session in which the call was made, call memory is said to be a "shorter-duration" memory relative to the session memory. Conversely, session memory is "longer-duration" memory relative to call memory. (Specification, p. 10, line 9 - p. 11, line 23, Fig. 2)

A run-time external reference (XREF) is a reference to an object, such that dereferencing the run-time external reference causes the external object to be loaded into virtual memory, if the object is not currently loaded. (Specification, p. 13, lines 13-17)

When XREFs to objects in shorter-duration memory are used in longer-duration memory, it results in a significant performance penalty because the objects in the longer-duration memory do not contain information specific to any particular call. However, because the recreated objects are realized on a per-call basis, the XREF pointers array may reside in the call memory, which is private. Consequently, XREFs in longer-duration memory do not contain pointers to the corresponding XREF pointers arrays. However, the XREF pointers array that corresponds to an XREF must be inspected every time the XREF is dereferenced in order to determine whether the referenced object has already been realized, and if so, where the realized instance of the object is located. Thus, every time an XREF in longer-duration memory is dereferenced, relatively expensive measures must be taken to locate the appropriate private XREF pointers array to determine whether the referenced object has already been realized.

Unfortunately, in systems that use lazy evaluation for resolving recreatable external references (e.g., deferring the allocation of external objects in session memory until those external objects are actually accessed), the runtime performance penalty for dereferencing such external references may be significant. Based on the foregoing, it is desirable to provide a system that allows lazy evaluation of external references, but reduces the runtime performance penalty. (Specification, p. 19, lines 8-23; p. 5, lines 4-5)

According to one aspect, pointers to XREF pointers arrays are cached in a context structure, where the context structure is passed as an argument when a call is made. Specifically, when a call is made, an array is allocated in the context structure that is passed as a parameter to the call. Initially, the array will not contain any populated entries. When an XREF is

encountered on a shared-memory page, a hash code of the page is located and the hash code (or some subset thereof), is extracted. However, rather than making a function call to locate a hash table entry, the portion of the hash code is used as an index into the array within the context structure. According to one aspect, the size of the array is 2^N, and the number of bits of the hash code that are used to index into the array is N. Thus, every possible index value will have a corresponding slot within the array.

If the array contains no entry at the indexed location, then an XREF pointers array has not yet been constructed for the shared-memory page during that call. Under those conditions, (1) an XREF pointers array is constructed, (2) an array entry that points to the new XREF pointers array is added to the array, (3) the object associated with the XREF is activated based on the corresponding descriptor array entry, and (4) an entry is added to the new XREF pointers array to point to the newly activated object.

If the array contains an entry at the indexed location, then it is determined whether the entry is for the correct shared-memory page. This step is performed because the array may be relatively small, and collisions may occur (i.e. different shared-memory pages may map to the same index value). According to one aspect, the key of a shared-memory page is stored within the entry for the shared memory page. Collisions may be identified by comparing the key stored in an entry with the key of the shared-memory page that is the subject of the search. In the case of a collision, any one of a number of well-known collision handling techniques may be employed. The present invention is not limited to any particular collision handling technique. In general, upon identifying a collision, a search for the correct entry will continue until either the correct entry is found, or it is determined that the array does not contain an entry for the correct

shared-memory page. If the array does not contain an entry for the correct shared-memory page, then the process described above is followed.

If a correct entry is identified, then the XREF pointers array associated with the entry is located based on a pointer within the array entry. The XREF pointers array is searched to determine whether the object associated with the XREF has been activated. If the object associated with the XREF has not been activated, then (1) the object associated with the XREF is activated based on the corresponding descriptor array entry, and (2) an entry is added to the new XREF pointers array to point to the newly activated object.

If the object associated with the XREF has already been activated, then a pointer in the appropriate entry of the XREF pointers array is used to locate the activated instance of the object. (Specification, p. 21, line 15 - p. 23, line 3)

Another type of cache yields good results when the XREFs are to constant pool variable values. Specifically, the code referencing the constant pool maintains a "session-private" version of each class (referred to herein as a "jom_active_class object") in addition to the shared version of that class (referred to herein as the "jom_class"). According to one aspect, the session-private recreatable external reference state for an object memory is stored in the jom_active_class for that object memory. Because there is only one class in each shared object memory that holds class information, the association between classes and the object memories that contain them is 1-1. Thus, when an XREF encountered in an object memory stored in longer-duration memory is a reference to a constant pool item, the jom_active_class object associated with the class in that object memory is inspected.

If the jom_active_class object contains no pointer to an XREF pointers array, then an XREF pointers array has not yet been constructed for the corresponding jom_class during that call. Under those conditions, (1) an XREF pointers array is constructed, (2) a pointer that points

to the new XREF pointers array is added to the jom_active_class object, (3) the object associated with the XREF is activated based on the corresponding descriptor array entry, and (4) an entry is added to the new XREF pointers array to point to the newly activated object.

If the jom_active_class object contains a pointer to the XREF pointers array, then the XREF pointers array associated with the class is located based on the pointer. The XREF pointers array is searched to determine whether the object associated with the XREF has been activated. If the object associated with the XREF has not been activated, then (1) the object associated with the XREF is activated based on the corresponding descriptor array entry, and (2) an entry is added to the new XREF pointers array to point to the newly activated object.

If the object associated with the XREF has already been activated, then a pointer in the appropriate entry of the XREF pointers array is used to locate the activated instance of the object.

In an alternative aspect, the XREF pointers array for a class is automatically built in call memory at the time that the class is activated. Thus, even the first dereferencing operation will encounter a pointer, within the jom_active_class object, to an XREF pointers array. (Specification, p. 23, line 10 - p. 24, line 15)

VI. <u>GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL</u>

Whether claims 1, 4, 6-10, 13, and 15-20 are obvious under 35 U.S.C. § 103(a) as obvious over *Chang et al.* (US 5,870,753).

Whether claims 2-3 and 11-12 under 35 U.S.C. § 103(a) are obvious over *Chang et al.* in view of *Bennett* (US 6,014,733).

Whether claims 5 and 14 under 35 U.S.C. § 103(a) are obvious over *Chang et al.* in view of *Bennett* and further in view of *Jones et al.* (US 6,629,154).

VII. ARGUMENT

A. CLAIMS 1-20 ARE NOT RENDERED OBVIOUS BY *CHANG ET AL.* OR IN VIEW OF *BENNETT* OR FURTHER IN VIEW OF *JONES ET AL.*

The initial burden of establishing a *prima facie* basis to deny patentability to a claimed invention under any statutory provision always rests upon the Examiner. *In re Mayne*, 104 F.3d 1339, 41 USPQ2d 1451 (Fed .Cir. 1997); *In re Deuel*, 51 F.3d 1552, 34 USPQ2d 1210 (Fed. Cir. 1995); *In re Bell*, 991 F.2d 781, 26 USPQ2d 1529 (Fed. Cir. 1993); *In re Oetiker*, 977 F.2d 1443, 24 USPQ2d 1443 (Fed. Cir. 1992). In rejecting a claim under 35 U.S.C. § 103, the Examiner is required to provide a factual basis to support the obviousness conclusion. *In re Warner*, 379 F.2d 1011, 154 USPQ 173 (CCPA 1967); *In re Lunsford*, 357 F.2d 385, 148 USPQ 721 (CCPA 1966); *In re Freed*, 425 F.2d 785, 165 USPQ 570 (CCPA 1970).

Obviousness rejections require some evidence in the prior art of a teaching, motivation, or suggestion to combine and modify the prior art references. See, e.g., *McGinley v. Franklin Sports, Inc.*, 262 F.3d 1339, 1351-52, 60 USPQ2d 1001, 1008 (Fed. Cir. 2001); *Brown & Williamson Tobacco Corp. v. Philip Morris Inc.*, 229 F.3d 1120, 1124-25, 56 USPQ2d 1456, 1459 (Fed. Cir. 2000); *In re Dembiczak*, 175 F.3d 994, 999, 50 USPQ2d 1614, 1617 (Fed. Cir. 1999). The Patent Office must give specific reasons why one of ordinary skill in the art would have been motivated to combine the references. See, e.g., *In re Kotzab*, 217 F.3d 1365, 1371, 55 USPQ2d 1313, 1317 (Fed. Cir. 2000); *In re Rouffet*, 149 F.3d 1350, 1359, 47 USPQ2d 1453, 1459 (Fed. Cir. 1998).

Reversal of this rejection is respectfully requested because *Chang et al.* alone (or even in combination with *Bennett* and *Jones et al.*) does not disclose, teach, or otherwise suggest the features of the claims, and there is no legally permissible motivation to modify *Chang et al.* to reach the claim limitations. For example, independent claims 1 and 10 recite:

> if the XREF pointers array includes a pointer associated with said reference located in the longer-duration memory, then following said pointer to locate said instance within said shorter-duration memory.

As another example, independent claims 6 and 15 recite:

> to dereference said reference located in the longer-duration memory.

These features are not shown in *Chang et al.*, as the Office Action correctly acknowledged: "Chang does not explicitly teach the pointer associated with the reference located in the longer-duration memory" (Office Action, page 3, ¶ 3; page 4, ¶ 10), and reacknowledged in the Advisory Action: "Chang does not teach the pointer associated with the reference located in the longer-duration memory" (Advisory Action, p. 2). In fact, *Chang et al.* fails to show any reference located in persistent storage **74'** at all. *Chang et al.* at best shows the opposite direction: the object reference **76** located in server memory **70** points to state **73'** in persistent storage **74'**, not the other way round.

Despite *Chang et al.*'s lack of evidentiary support for the rejection, the Examiner contended it would have been obvious "to improve the system of Chang because it would improve the performance of the system when determining the existing [*sic*, existence?] of the object" (Office Action, p. 3, ¶ 7). There is nothing in the record, however, indicating how the proposed modification would actually "improve the performance of the system." In fact, the disclosure of *Chang et al.* contradicts the Examiner's contention, because access of the key **78** in persistent storage **92** of *Chang et al.* would add unnecessary processing time to the already disclosed access of the key **78** in the memory **90** to find the pointer **77** in the reference data table in memory **90**. If a proposed modification would render the prior art being modified unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed modification. *In re Gordon*, 733 F.2d 900, 221 USPQ 1125 (Fed. Cir. 1984).

In support of the obviousness rejection, the Examiner also stated that "the shorter-duration memory associated with the reference [is?] located in the shorter-duration memory and longer-duration memory (col. 5, lines 23-59 and Figs. 8-9)" (*id.*). The Examiner's statement is somewhat confusing because it implies that object reference **76** can exist in two different places at the same time, whereas *Chang et al.* only discloses it to be located in server memory **70.**[1] The cited passage of *Chang et al.* merely states that copies of one part of the object reference **76** (that is, the key **78**) are stored in both (RAM) memory and persistent storage: "Object reference **76,** contains the key **78**, which is used by the server process to associate the object reference with a row in the reference data table in both persistent storage **92** and in memory **90**" (col. 5:49-51). Accordingly, the record lacks the requisite factual basis to sustain the rejection.

The Office Action also argued: "It is also noted that **in the specification**, the XREF pointers array is searched to determine whether the object associated with the XREF has been activated" (Office Action, p. ¶ 6, emphasis added). Obviousness rejections, however, require some evidence **in the prior art** of a teaching, motivation, or suggestion to combine and modify the prior art references. *See, e.g., McGinley v. Franklin Sports, Inc.*, 262 F.3d 1339, 1351-52, 60 USPQ2d 1001, 1008 (Fed. Cir. 2001); *Brown & Williamson Tobacco Corp. v. Philip Morris Inc.*, 229 F.3d 1120, 1124-25, 56 USPQ2d 1456, 1459 (Fed. Cir. 2000); *In re Dembiczak*, 175 F.3d 994, 999, 50 USPQ2d 1614, 1617 (Fed. Cir. 1999). Here, the explicit use of portions of the specification of the present application for suggestions to modify *Chang et al.* indicates that the motivation is influenced by impermissible hindsight.

[1] The Advisory Action makes the same confusing contention with respect to pointer **77**: "Chang actually teaches (col. 5, lines 23-59 and Fig. 9) the pointer associated with the reference (object's metadata 71) located in the shorter-duration memory (server process memory) and also located in the longer-duration memory (relational database/persistent storage)."

Furthermore, the proposed modification to *Chang et al.* requires non-analogous art. "A reference is reasonably pertinent if, even though it may be in a different field from that of the inventor's endeavor, it is one which, because of the matter with which it deals, logically would have commended itself to an inventor's attention in considering his problem." *In re Clay*, 966 F.2d 656 (Fed. Cir. 1992). *Chang et al.* is not reasonably pertinent to the problem with which the present inventors are concerned, namely "accessing an instance of a recreatable object in a shorter-duration memory based on a reference located in a longer-duration memory" (claim 1). The problem that *Chang et al.* is concerned with is the problem of maintenance of multiple metastates for a persistent object without increasing the size of the object reference (Abstract). *Chang et al.* is blissfully unaware of any problems with accessing shorter-duration instances of activatable objects based on object references stored in longer-duration memory. Because *Chang et al.* is not reasonably pertinent to addressing issues with object references stored in longer-duration memory, there is no motivation for one of ordinary skill in the art to modify the disclosure of *Chang et al.*

As a further example, dependent claims 7 and 16 recite:

> the step of storing, within said class object, data for locating instances is performed by **storing, within said class object, a pointer to an XREF pointers array**

The Office Action stated that this feature is taught by *Chang et al.* at col. 12, lines 53-59. (Office Action, p. 4, ¶ 12) However, the cited portion of *Chang et al.* merely mentions invoking a generic initializer on an object's class to create a new object, and makes no mention or suggestion of "storing, within said class object, a pointer to an XREF pointers array." Furthermore, there is no motivation given by the Office Action for modifying *Chang et al.* to include such a feature. Obviousness rejections, however, require some evidence in the prior art

of a teaching, motivation, or suggestion to combine and modify the prior art references. *See, e.g., McGinley v. Franklin Sports, Inc., supra.* Moreover, dependent claims 8 and 17, which depend from claims 7 and 16, respectively, recite:

> determining whether the XREF pointers array includes a pointer associated with said reference

which the Office Action contends is taught by *Chang et al.* at col. 5, lines 25-30. (Office Action, p. 4, ¶ 13) However, this cited portion of *Chang et al.* refers to an object in memory **72** of **Fig. 8**, which is clearly labeled in **Fig. 8** as "persistent object," which is inconsistent with "generating, within shorter-duration memory, a class object" and the "storing" step recited at least by claims 8 and 17. Appellants assert that the reasoning that the Examiner puts forth for the rejection with respect to "determining whether the XREF pointers array includes a pointer associated with said reference" contravenes 35 U.S.C. § 132, which requires the Director to "notify the applicant thereof, stating the reasons for such rejection." This section is violated if the rejection "is so uninformative that it prevents the applicant from recognizing and seeking to counter the grounds for rejection." *Chester v. Miller*, 906 F.2d 1574, 15 USPQ2d 1333 (Fed. Cir. 1990). This policy is captured in the Manual of Patent Examining Procedure. For example, MPEP § 706 states that "[t]he goal of examination is to clearly articulate any rejection early in the prosecution process so that applicant has the opportunity to provide evidence of patentability and otherwise respond completely at the earliest opportunity." Furthermore, MPEP § 706.02(j) indicates that: "[i]t is important for an examiner to properly communicate the basis for a rejection so that the issues can be identified early and the applicant can be given fair opportunity to reply." Here, the Examiner has cited a portion of *Chang et al.* that is inconsistent with the clearly recited claim features, and thus has not properly communicated a basis for the rejection.

Bennett, applied for the rejection of claims 2-3 and 11-12, is directed to a method for creating a "perfect hash" and also does not show the above-quoted limitations of the claims. Further, *Jones et al.*, applied only to claims 5 and 14, is directed to a method for identifying a remote method to invoke on a server using a deterministic hash and also does not show the above-quoted limitations of the claims, either alone or in any combination with *Chang et al.* and *Bennett*. Therefore, the rejection of claims 1-20 should be withdrawn.

VIII. CONCLUSION AND PRAYER FOR RELIEF

For the foregoing reasons, Appellants request the Honorable Board to reverse each of the Examiner's rejections.

Respectfully Submitted,

DITTHAVONG & CARLSON, P.C.

November 19, 2004
Date

Margo Livesay
Stephen C. Carlson
Attorney for Applicant(s)
Reg. No. 39,929

10507 Braddock Rd, Suite A
Fairfax, VA 22032
Tel. 703-425-8516
Fax. 703-425-8518

Margo Livesay, Ph.D.
Reg. No. 41,946

APPENDIX

1. (Previously Presented) A method for accessing an instance of a recreatable object in a shorter-duration memory based on a reference located in a longer-duration memory, wherein the shorter-duration memory is associated with a call, the method comprising the steps of:

locating, within the shorter-duration memory, a context structure associated with the call;

locating an XREF pointers array based on data cached within the context structure;

determining whether the XREF pointers array includes a pointer associated with said reference located in the longer-duration memory; and

if the XREF pointers array includes a pointer associated with said reference located in the longer-duration memory, then following said pointer to locate said instance within said shorter-duration memory.

2. (Previously Presented) The method of Claim 1 wherein the step of locating an XREF pointers array based on data cached within the context structure comprises the steps of:

determining a hash code associated with a memory page in which the reference located in the longer-duration memory is located;

using at least a portion of the hash code as an index to locate an array entry within an array stored within the context structure; and

if said array entry contains a pointer, then following said pointer stored in said array entry to locate said XREF pointers array.

3. (Original) The method of Claim 2 wherein:

the array is a power-of-two array; and

the portion of said hash code that is used as said index includes a particular number of bits of said hash code.

4. (Original) The method of Claim 1 wherein:

the XREF pointers array does not include a pointer associated with said reference; and

the method further comprises the steps of

creating said instance by activating said recreatable object; and

storing a pointer to said instance in said XREF pointers array.

5. (Original) The method of Claim 2 wherein:

if said array entry does not contain a pointer, then creating said instance by activating said recreatable object; and

storing a pointer to said instance in said array entry.

6. (Previously Presented) A method for accessing an instance of a recreatable object in shorter-duration memory based on a reference located in a longer-duration memory, wherein the shorter-duration memory is associated with a call, the method comprising the steps of:

when a class is activated, generating, within said shorter-duration memory, a class object associated with the class;

storing, within said class object, data for locating instances of recreatable objects associated with said class;

to dereference said reference located in the longer-duration memory, performing the steps of

determining that said reference located in a longer-duration memory is associated with said class; and

using said data within said class object to locate said instance of said recreatable object.

7. (Original) The method of Claim 6 wherein the step of storing, within said class object, data for locating instances is performed by storing, within said class object, a pointer to an XREF pointers array.

8. (Original) The method of Claim 7 wherein the step of using said data within object to locate said instance includes the steps of:

determining whether the XREF pointers array includes a pointer associated with said reference;

if the XREF pointers array includes a pointer associated with said reference, then following said pointer to locate said instance within said shorter-duration memory.

9. (Original) The method of Claim 8 wherein:

the XREF pointers array does not include a pointer associated with said reference; and

the method further comprises the steps of

creating said instance by activating said recreatable object; and

storing a pointer to said instance in said XREF pointers array.

10. (Previously Presented) A computer-readable medium carrying instructions for accessing an instance of a recreatable object in a shorter-duration memory based on a reference located in a longer-duration memory, wherein the shorter-duration memory is associated with a call, the computer-readable medium comprising instructions for performing the steps of:

locating, within the shorter-duration memory, a context structure associated with the call;

locating an XREF pointers array based on data cached within the context structure;

determining whether the XREF pointers array includes a pointer associated with said reference located in the longer-duration memory; and

if the XREF pointers array includes a pointer associated with said reference located in the longer-duration memory, then following said pointer to locate said instance within said shorter-duration memory.

11. (Previously Presented) The computer-readable medium of Claim 10 wherein the step of locating an XREF pointers array based on data cached within the context structure comprises the steps of:

determining a hash code associated with a memory page in which the reference located in the longer-duration memory is located;

using at least a portion of the hash code as an index to locate an array entry within an array stored within the context structure; and

if said array entry contains a pointer, then following said pointer stored in said array entry to locate said XREF pointers array.

12. (Original) The computer-readable medium of Claim 11 wherein:

the array is a power-of-two array; and

the portion of said hash code that is used as said index includes a particular number of bits of said hash code.

13. (Original) The computer-readable medium of Claim 10 wherein:

the XREF pointers array does not include a pointer associated with said reference; and

the computer-readable medium further comprises instructions for performing the steps of

creating said instance by activating said recreatable object; and

storing a pointer to said instance in said XREF pointers array.

14. (Original) The computer-readable medium of Claim 11 further comprising instructions for performing the steps of:

if said array entry does not contain a pointer, then creating said instance by activating said recreatable object; and

storing a pointer to said instance in said array entry.

15. (Previously Presented) A computer-readable medium carrying instructions for accessing an instance of a recreatable object in shorter-duration memory based on a reference located in a longer-duration memory, wherein the shorter-duration memory is associated with a call, the computer-readable medium comprising instructions for performing the steps of:

when a class is activated, generating, within said shorter-duration memory, a class object associated with the class;

storing, within said class object, data for locating instances of recreatable objects associated with said class;

to dereference said reference located in the longer-duration memory, performing the steps of

determining that said reference located in the longer-duration memory is associated with said class; and

using said data within said class object to locate said instance of said recreatable object.

16. (Original) The computer-readable medium of Claim 15 wherein the step of storing, within said class object, data for locating instances is performed by storing, within said class object, a pointer to an XREF pointers array.

17. (Original) The computer-readable medium of Claim 16 wherein the step of using said data within object to locate said instance includes the steps of:

determining whether the XREF pointers array includes a pointer associated with said reference;

if the XREF pointers array includes a pointer associated with said reference, then following said pointer to locate said instance within said shorter-duration memory.

18. (Original) The computer-readable medium of Claim 17 wherein:

the XREF pointers array does not include a pointer associated with said reference; and

the computer-readable medium further comprises instructions for performing the steps of

creating said instance by activating said recreatable object; and

storing a pointer to said instance in said XREF pointers array.

19. (Previously Presented) The method of Claim 1 wherein the duration of the shorter-duration memory is shorter than the duration of the longer-duration memory.

20. (Previously Presented) The computer-readable medium of Claim 10 wherein the duration of the shorter-duration memory is shorter than the duration of the longer-duration memory.

AF/2126 IFW

PTO/SB/21 (12-97)
Approved for use through 9/30/00. OMB 0651-0031
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	09/768,823
Filing Date	January 25, 2001
In re Application of:	Harlan SEXTON et al.
Group Art Unit	2126
Examiner Name	Cao, D.
Attorney Docket Number	50277-0459
Client Docket Number	OID-1999-084-01
Total Number of Pages in This Submission	23

ENCLOSURES (check all that apply)

- [x] Fee Transmittal Form
  - [x] Fee Attached
- [ ] Amendment / Response
  - [ ] After Final
  - [ ] Affidavits/declaration(s)
- [ ] Extension of Time Request
- [ ] Express Abandonment Request
- [ ] Information Disclosure Statement
- [ ] Certified Copy of Priority Document(s)
- [ ] Response to Missing Parts/ Incomplete Application
  - [ ] Response to Missing Parts under 37 CFR 1.52 or 1.53
- [ ] Assignment Papers (for an Application)
- [ ] Drawing(s)
- [ ] Licensing-related Papers
- [ ] Petition Routing Slip (PTO/SB/69) and Accompanying Petition
- [ ] To Convert a Provisional Application
- [ ] Power of Attorney, Revocation Change of Correspondence Address
- [ ] Terminal Disclaimer
- [ ] Small Entity Statement
- [ ] Request of Refund
- [ ] After Allowance Communication to Group
- [ ] Appeal Communication to Board of Appeals and Interferences
- [x] Appeal Communication to Group (Appeal Notice, Brief, Reply Brief)
- [ ] Proprietary Information
- [ ] Status Letter
- [ ] Additional Enclosure(s) (please identify below):

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm or Individual name	DITTHAVONG & CARLSON, P.C. Margo Livesay, Ph.D. Reg No. 41946
Signature	Margo Livesay
Date	November 19, 2004

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Alexandria, VA 22313-1450 on this date:

Type or printed name	Linda V. Wiley		
Signature		Date	November 19, 2004